SDK QUICKSTART

Set up an Eclipse Environment for uDig Plug-in Development

27 June 2008





TABLE OF CONTENTS

1 Goals	3
2 Downloads	4
3 Eclipse SDK Installation	5
4 Eclipse Workspace	7
5 Eclipse Preferences	8
6 uDIG SDK	9
7 Running uDig	11
8 What to Do Next	13

1 GOALS

This Workbook is aimed at those doing plug-in development against the UDIG platform. Follow these instructions to quickly set up a development environment for working on your own plug-ins.

Eclipse is familiar to most developers as a Java Integrated Development Environment (IDE). The Eclipse IDE can be extended with additional "capabilities" to work with alternate programming languages (like C++ or Ruby), or additional subject matter such as Java Enterprise Edition or in this case Eclipse Plug-in development.

In this Quickstart we are going to use the the Eclipse Plug-in Development capability; with the uDIG SDK as the target platform.

This workbook covers setting up a development environment for working on your own plug-ins.

If you have an existing Eclipse installation please do not skip this tutorial – we are going to very carefully set up a copy of Eclipse with a few more additional tools then you are used to.

2 DOWNLOADS

If you are using this work book in a lab setting you will find these downloads available on your DVD: D:\udig\downloads

The extras pack contains language files for the RCP-base portions of uDig, for German, Spanish, French, Italian, Japanese, Korean, Portuguese (Brazil), Traditional Chinese and Simplified Chinese. We are going to start by downloading all the software we need; we will be able to proceed with installation as we wait for some of the larger downloads.

1. Visit the uDig website for the latest UDIG SDK: <u>http://udig.refractions.net/downloads</u>

At the time of writing this file was: <u>udig-1.1-RC14-sdk.zip</u>

2. Visit <u>http://www.eclipse.org/downloads</u> and click on the link: "Eclipse for RCP/Plug-in Developers"

Tested with Eclipse 3.3.2 for RCP/Plug-in Developers: eclipse-rcp-europa-winter-win32.zip eclipse-rcp-europa-winter-linux-gtk.tar.gz eclipse-rcp-europa-winter-macosx-carbon.tar.gz

3. We have prepared an "extras" download in the following folder: http://udig.refractions.net/downloads/extras/

This download includes source for EMF, GEF along with a Developers guide, and Platform language packs. Please download the file with a version number matching your eclipse.

At the time of writing: extras-3.3.2.zip

4. Download a Java Runtime Environment from this folder: <u>http://udig.refractions.net/downloads/jre/</u>

This is a special JRE that has been extended with Java Advanced Imaging and Image IO.

At the time of writing the following was available: jre1.6.0_06.win32.zip jre1.5.0_08.linux.zip

3 ECLIPSE SDK INSTALLATION

Hopefully by now your eclipse download has finished and we can begin to installation.

- 1. Create a folder: <u>C:∖java</u>
- 2. Unzip the downloaded eclipse **RCP/Plug-in Developers** file to your java directory:

C:\java\eclipse will be created



Extract the jre zip file into your eclipse directory.
 C:\java\eclipse\jre will be created.



4. Unzip the **extras-3.3.2.zip** file to your eclipse directory. The download will add additional plug-ins and features to to your eclipse directory.

Depending on your unzip program you may be asked to "merge folders" or "replace files".

If you need a good program to unzip archive files try: <u>http://www.7-zip.org/</u>

The folder must be called "jre" for the eclipse.exe to find it automatically.

The "extras" download contains a "plugins" and "features" folder.

- If you are on a linux or mac osx machine you may want to modify the "eclipse.ini" file to specify additional command line options.
- 5. Navigate to C:\java\eclipse and right-click on the eclipse.exe file and select Send To->Desktop (create shortcut).
- 6. Open up the desktop short cut properties and change the Target: C:\java\eclipse.exe -vmargs -Xmx512m

Shortcut to ecl	lipse.exe Properties
General Shortc	ut Compatibility
sł	nortcut to eclipse.exe
Target type:	Application
Target location:	eclipse
Target:	C:\java\eclipse\eclipse.exe -vmargs -Xmx512m
Start in:	C:\java\eclipse
Shortcut key:	None
<u>R</u> un:	Normal window
Comment:	
<u>F</u> ind	Target Change Icon Advanced
	OK Cancel Apply

4 ECLIPSE WORKSPACE

The Eclipse IDE keeps track of what you are doing in a workspace. You can have several workspaces (often for different projects) each with its own configuration.

- 1. Double click on your desktop short cut to start up eclipse. When you start up eclipse for the first time it prompt you for a workspace.
- 2. Choose a workspace for your sdk development: <u>C:\java\workspace</u>



- 3. Wait a few moments while eclipse starts up.
- 4. On the Welcome view press the **Workbench** button along the right hand side.



5 ECLIPSE PREFERENCES

We have a few global settings to configure before we can proceed.

- 1. Open up **Window > Preferences** from the menu bar.
- 2. Navigate to the **Java > Compiler** page and change:
 - Compiler compliance level: 5.0
- 3. Check the Java > Installed JREs page:
 - Should have Location: C:\java\eclipse\jre

If the setting is not correct you can use the **Add.** button and create a JRE entry for C:\java\eclipse\jre

• You can press Edit to look at the installed JRE.

Regardless of platform we are interested in making sure jai core.jar, jai imageio.jar and jai codec.jar are available.

-		
JRE <u>t</u> ype:	Standard VM 👻	
JRE <u>n</u> ame:	jre	
JRE home directory:	C:\java\eclipse\jre	<u>B</u> rowse
Default <u>V</u> M Arguments:		
JRE system libraries:		
C:\java\eclipse	\jre\lib\rt.jar	Add External JARs
G:\java\eclipse C:\java\eclipse	\jre\lib\jsse.jar \jre\lib\jce.jar	Javadoc Location
C:\java\eclipse\jre\lib\charsets.jar		
C:\java\eclipse	\jre\lib\ext\dnsns.jar	Re <u>m</u> ove
C:\java\eclipse C:\java\eclipse	\jre\lib\ext\jai_codec.jar \jre\lib\ext\jai_core.jar	Up
C:\java\eclipse	\jre\lib\ext\jai_imageio.jar	Down
C:\java\eclipse	\jre\lib\ext\mlibwrapper_jai.jar	Restore Default
C:\java\eclipse	\jre\lib\ext\sunjce_provider.jar \ire\lib\ext\sunmscani jar	
C:\java\eclipse	\jre\lib\ext\sunpkcs11.jar	
L		
⑦ OK Cancel		

We are waiting for a Mac OSX JRE to be available before using Java 6.

On OSX we are not able to provide a JRE for you to download – you will need to download and install JAI and ImageIO yourself.

6 UDIG SDK

We are now going to unpack the udig sdk and use it as our plug-in target platform.

1. Extract the sdk download into C:\java\sdk\



- 2. Go back to eclipse and open Window>Preferences.
- 3. Select the **Plugin Development > Target Platform** page.
- Use the Browse... button to change the Location to: C:\java\sdk\udig



5. The list of plug-ins for the uDig application

Preferences	
type filter text	Target Platform $\Leftrightarrow \circ \Rightarrow \circ$
type filter text General Ant Cache Install/Update Java Mylyn Complers Complers Complers CosGi Frameworks Target Platform Run/Debug Team Validation Web and XML	Target Platform Specify the platform against which the workspace plug-ins will be compiled and tested: Location: C:ljavalgdk/udig-dk Plug-ins Environment Lug-ins Eleoad Locations Eleoad Locations.udig.catalog.arcalog.arcatolog.tin.10) Eleoad
	Restore Defaults Apply
0	OK Cancel

6. Press OK

At this point all the source code for the Eclipse and uDig plug-ins are available. We can now start working on uDig plug-ins, but before we do that lets try running the application.

7 RUNNING UDIG

With all this in place we can now run the uDig application from your development environment. This is a good way to test that everything is installed correctly.

- 1. Select Run/Open Run Dialog... from the menu bar
- 2. Select Eclipse Application from the list of configuration types
- 3. Press the New button, and confirm the following details:
 - Ensure that **Run a Product** is selected (Under Program to Run section on right).
 - Change the **Run a Product** field to: net.refractions.udig.product

🥏 Run	
Create, manage, and run configu Create a configuration to launch an Eclipse	a application.
Image: Second Secon	Name: New_configuration Main Me* Arguments Plug-ins Configuration Tracing *2 Workspace Data Location: \${workspace_loc}//runtime+New_configuration Image: Configuration Image:
Filter matched 9 of 9 items	Appi <u>y</u> Re <u>v</u> ert
0	<u>R</u> un Close

4. Switch to the **Arguments** tab and set the **VMArguments** to: -Xmx512m -Dosgi.parentClassloader=ext

Create a configuration to launch an Eclip	se application.
Image: Second system Image: Second system Image: Image: Image: Second system Image: Second system Image: Image: Image: Image: Second system Image: Second system Image:	Name: New_configuration Main Main Program arguments: Tracing -os \${target.os} -ws \${target.ws} -arch \${target.arch} -nl \${target.nl} Variables VM arguments: -Xmx512m -Dosgi.parentClassloader =ext Variables Working directory: O Default: C: \ava\ecipse Other: Workspace Ele System Variables
ilter matched 9 of 9 items	Apply Reyert

5. Press **Run** to start the application.



You may notice a few tools and options not normally included with the uDig application.

The SDK includes these extra code examples for you to learn from.

8 WHAT TO DO NEXT

Here are some additional things to try when running uDig. Please take a moment to read over the following suggestions – you will use these tricks when debugging.

• Add -consoleLog to your "program arguments" to send log information to standard out. This allows you to watch the tracing information on the console as uDig runs.

🔄 <u>M</u> ain 🕺 Arguments 🔪 🧏 Plug-ins 🔛 Configura <u>t</u> ion 🚔 Tracing 🚾 Environme	nt 🔲 <u>C</u> ommon
Program <u>a</u> rguments:	
-os \${target.os} -ws \${target.ws} -arch \${target.arch} -nl \${target.nl:tar} -consoleLog	
	Variable
VM arguments:	
-Xmx386M -Dosgi,parentClassloader=ext	
	Variable

The running uDig application makes use of the "Workspace Data" folder defined in the Run dialog. Try checking clear and you can simulate starting uDig from a fresh install.
 Main M-Arguments Plug-ins Configuration Tracing Environment Common



• Have a look on the Tracing tab of the Run dialog; turn on a few options.



• Have a look at the plug-ins tab and see if you can turn off: printing support.

Main 🖾 Arguments 🐝 Plug-ins 🔠 Confi	igura <u>t</u> ion 🛗 Irac <u>i</u> ng 🔪 🌆 Environment 🛄 <u>C</u> ommon
☑ Enable tracing	
met.refractions.udig.catalog.shp (1.1.0)	▲ debug
🔲 🕪 net.refractions.udig.catalog.wfs (1.1.0)	≡
Image: A state of the state	

Normally you turn off any plug-in with "test" or "tutorials" in the name; we will fix this up in the custom application tutorial.

For documentation on these command line parameters check the eclipse help menu.